All about CoAP

TP#3 using FIT/IoT-Lab

Lecture slides for RIO201

24-10-2018

TELECOM ParisTech

INSTITUT Mines-Télécom

Summary of previous TP

université PARIS-SACLAY

Public CoAP server tutorial

- **How?**
  - Install a CoAP server on a sensor node
  - Read the sensing data from front-end or local computer
- **Why?**
  - Get a feel of how data is transmitted using low-overhead protocols such as CoAP
  - Collect the data and do practically what you want
- **Before we go on...**
  - We will all do the tutorial again
  - (Future exercises all stem from this anyway)

Be careful!

- **Selecting your nodes**
  - Out of interference range of other nodes
  - Choose from Saclay, Lille, Grenoble

  - Let’s try to fix this before!

- **Border router**
  - Pick an ipv6 address within the range of subnet!

Example in python for CoAP

```
###################### Simple program for receiving CoAP data from Python
"""
import subprocess

command = "coap get "
coap_server = "coap://[2001:660:5307:3102::a881]"
port = ":5683"
output1 = "/sensors/accel"

string = command + coap_server + port + output1

result = subprocess.check_output(string, shell=True)

print(result)
"""
```

Example in python for HTTP

```
  GNU nano 2.2.6                Fichier : test.py

###################### Simple program for receiving HTTP data from Python
#"""
import subprocess

command = "lynx -dump "
http_server = "http://[2001:660:5307:3102::a881]"

string = command + http_server

result = subprocess.check_output(string, shell=True)

print(result)
#"""
```

Exercise

- **Create a python program that collects sensing data from a CoAP server every period**
  - Period = one second
  - Data = sensors/gyro, sensors/pressure
  - Node = 1 border router, 1 CoAP server

Tougher exercise?

- **Create a python program that collects sensing data from two CoAP servers every period**
  - Node = 1 border router, 2 CoAP servers
  - Period = one second per server #1, three seconds for server #2
  - Data = sensors/light
  - If light value is changed, printf an alarm message
- **How can we control two flow of data concurrently?**

Hint!

```
import time
from threading import Thread

def myfunc(i):
    print "sleeping 5 sec from thread %d" % i
    time.sleep(5)
    print "finished sleeping from thread %d" % i

for i in range(10):
    t = Thread(target=myfunc, args=(i,))
    t.start()
```


TELECOM
ParisTech
INSTITUT
Mines-Télécom

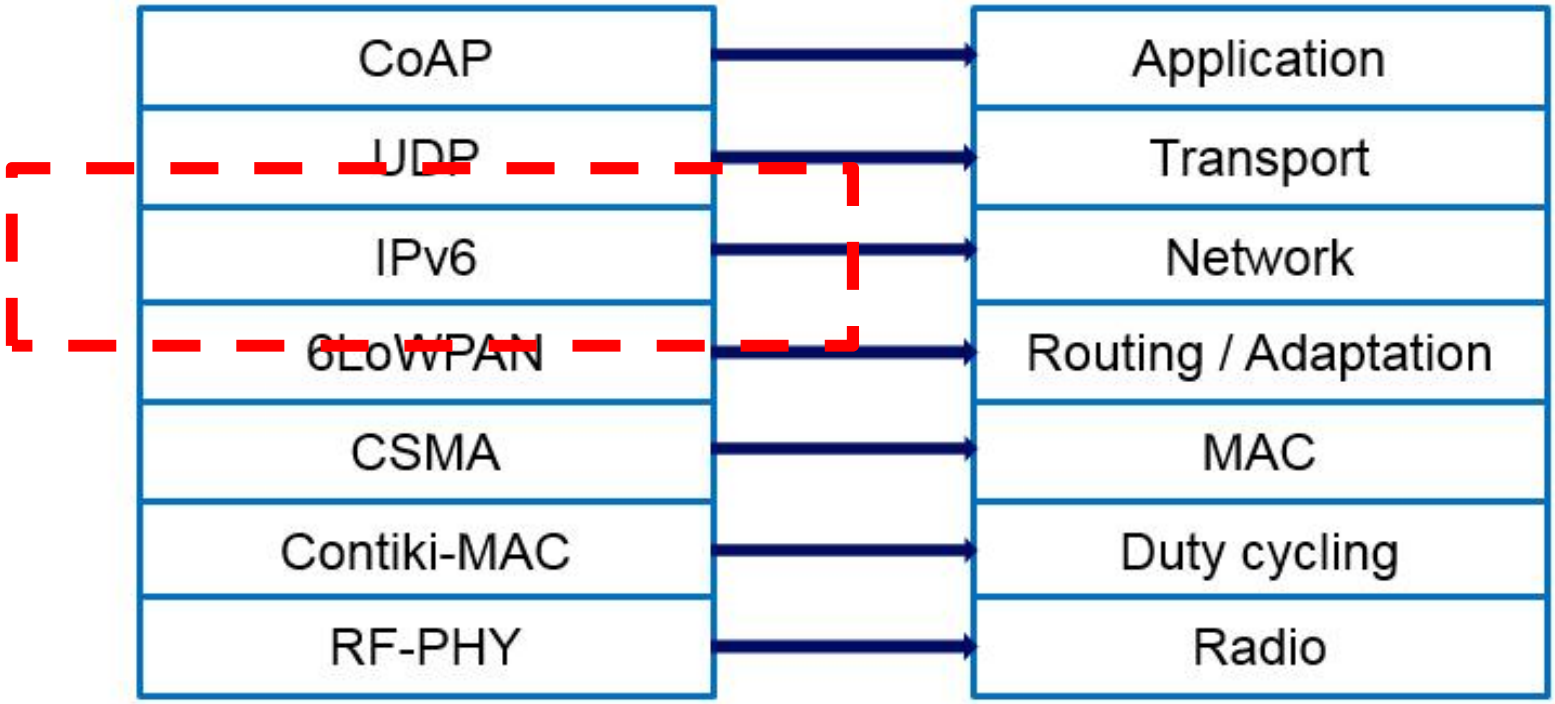
CoAP vs HTTP


université
PARIS-SACLAY

On why it is efficient to use COAP

- **We have been talking a lot about CoAP being better than HTTP**
  - But it would be better to see with our eyes
- **How to see this?**
  - Let's create a very simple packet sniffer
  - Pity we can't use TCPdump in FIT/IoT-Lab

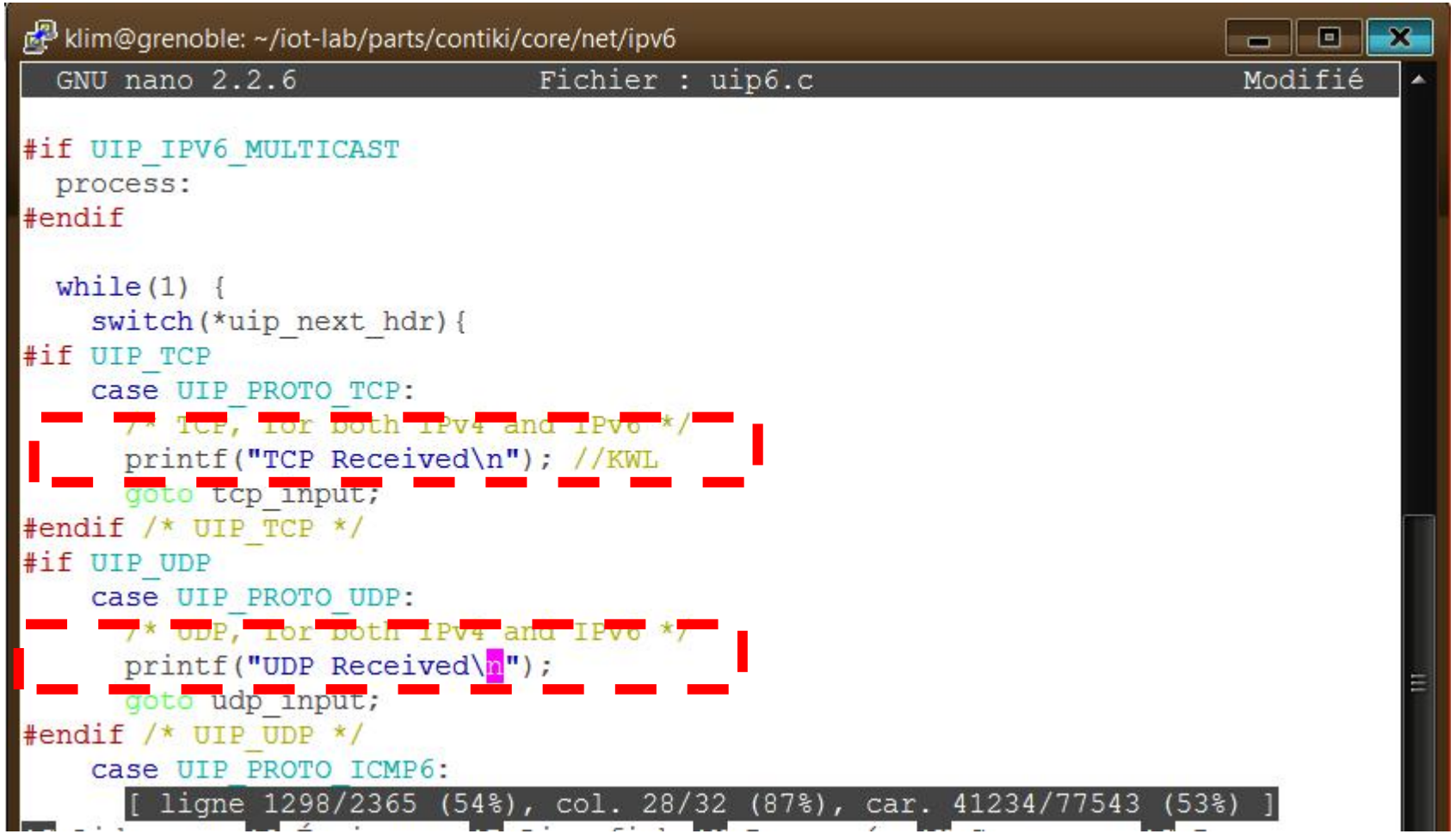
Where to sniff the packets?



Very simple sniffing code

- Goto

Part receiving TCP/UDP packets



Save and compile

- **Now the IPv6 will print data packets of CoAP (UDP) and HTTP (TCP) when they are received**

- **Go to compile coap-server**
  - Cd ~/iot-lab/parts/contiki/examples/iotlab/04(tab)
  - Make TARGET=iotlab-m3
  - Node-cli .....
  - Nc m3-XXX 20000

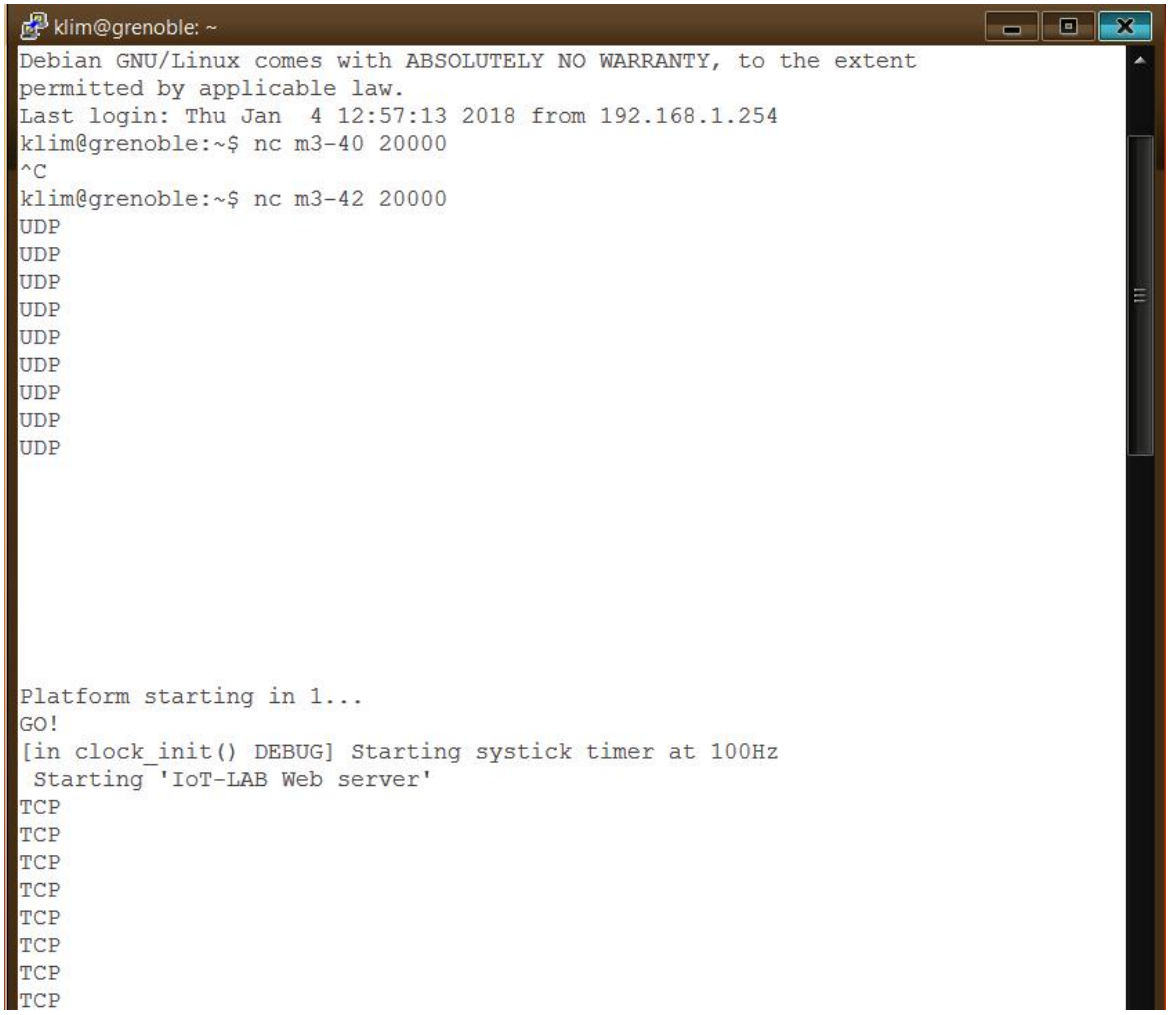
- **Send data in another shell**
  - Send a data through your python interface
  - Check output of UDP packets received

Save and compile HTTP

- **Go to compile http-server**
  - Cd ~/iot-lab/parts/contiki/examples/ipv6/http-server
  - Make TARGET=iotlab-m3
  - Node-cli .....
  - Nc m3-XXX 20000

- **Send data in another shell**
  - Send a data through your python interface
  - Check output of TCP received packets

What you should see



Balance HTTP and CoAP

- **Right now, CoAP only sends its data**
- **HTTP sends other information as well**

- **Try to balance the packet transmission**
  - For HTTP, remove all the parts of other transmission, and send only light sensor information
  - For CoAP send only light sensor information

- **What difference does this bring?**

TELECOM ParisTech

INSTITUT Mines-Télécom

Final project

université PARIS-SACLAY

HTTP vs CoAP

- **Make a performance evaluation between HTTP and CoAP**

- **Things to consider**
  - Create a server-client model which can use both HTTP and CoAP with multiple nodes
  - Utilize both HTTP and CoAP to make a performance evaluation
  - What to compare
    - Number of packet transmissions (Energy)
    - Delay (average time taken per request)
    - Transmission ratio (Reliability)

Assessment points

- **1) Make a survey on the comparison of HTTP vs CoAP**
  - What are the differences?
  - What do you expect as the performance?
- **2) Explain your code in detail**
  - All parts that you have changed for your project
- **3) Concluding point**
  - Does your performance evaluation reflect the survey that you have done?

Assessment points 2

- **4) Create different network architecture**
  - multiple nodes in a grid topology (1 hop from border to sensor)
  - Line topology from 1 ~ 5 nodes multi-hop (**Multi-hop!!!!!**)
  - What is the difference between these two topologies?
- **5) Number of transmissions**
  - How does the performance change with less transmission intervals (for example: 1 second ~ 50 milliseconds)
  - How does the performance change with number of multi-hop nodes? (e.g. 1 ~ 5 nodes multi-hop)
- **6) Delay**
  - How is delay affected according to increase in data traffic?
  - Use Python library to calculate timing
- **7) Transmission ratio**
  - Try to congest the network (increase number of hops, nodes, traffic, etc.)
  - See the performance between HTTP and CoAP
- **8) Basically, test HTTP and CoAP under various circumstances!**
  - Use your imagination to make various tests as you can
  - What other topologies can affect the network performance?
  - What other performance variables are there?

Assessment method

- **Submit a report by**
  - keunwoo.lim@telecom-paristech.fr
- **Due date**
  - 14 November 2018 (tentative)